**Εργαστήριο Μικροϋπολογιστών**

**5η εργαστηριακή άσκηση**

[Τμήμα: Β Ομάδα: 15](http://mycourses.ntua.gr/group/group.php)

Συνεργάτες: Μαρουφίδης Ιωάννης (03113506),

Περράκης Γεώργιος (03113511) ,

Σοφιανίδης Γεώργιος (03113179)

**1η Άσκηση**

.include "m16def.inc"

.def temp=r20

start:

ldi r24,0

out DDRB,r24 ; port B as input

ldi r24,LOW(RAMEND) ; initialize stack

out SPL,r24

ldi r25,HIGH(RAMEND)

out SPH,r25

ser r24

out DDRA,r24 ; port A as output

ldi temp,1

shift\_left:

out PORTA,temp

ldi r24,LOW(5)

ldi r25,HIGH(5)

rcall wait\_msec

in r21,PINB

sbic PINB,0x00 ; if PINB(0)=1 led stops moving

jmp shift\_left

rol temp

cpi temp,128

breq shift\_right

jmp shift\_left

shift\_right:

out PORTA,temp

ldi r24,LOW(5)

ldi r25,HIGH(5)

rcall wait\_msec

in r21,PINB

sbic PINB,0x00

jmp shift\_right

ror temp

cpi temp,1

breq shift\_left

jmp shift\_right

wait\_usec:

sbiw r24 ,1 ; 2 κύκλοι (0.250 μsec)

nop ; 1 κύκλος (0.125 μsec)

nop ; 1 κύκλος (0.125 μsec)

nop ; 1 κύκλος (0.125 μsec)

nop ; 1 κύκλος (0.125 μsec)

brne wait\_usec ; 1 ή 2 κύκλοι (0.125 ή 0.250 μsec)

ret ; 4 κύκλοι (0.500 μsec)

wait\_msec:

push r24 ; 2 κύκλοι (0.250 μsec)

push r25 ; 2 κύκλοι

ldi r24 , low(998) ; φόρτωσε τον καταχ. r25:r24 με 998 (1 κύκλος - 0.125 μsec)

ldi r25 , high(998) ; 1 κύκλος (0.125 μsec)

rcall wait\_usec ; 3 κύκλοι (0.375 μsec), προκαλεί συνολικά καθυστέρηση 998.375 μsec

pop r25 ; 2 κύκλοι (0.250 μsec)

pop r24 ; 2 κύκλοι

sbiw r24 , 1 ; 2 κύκλοι

brne wait\_msec ; 1 ή 2 κύκλοι (0.125 ή 0.250 μsec)

ret ; 4 κύκλοι (0.500 μsec)

Στην παραπάνω άσκηση υλοποιούμε την κίνηση ενός led στην θύρα εξόδου Α από αριστερά προς τα δεξιά και αντίστροφα, του οποίου η κίνηση να μπορεί να διακοπεί με βάση το LSB της θύρας εισόδου Β. Η καθυστέρηση στο άναμμα του led είναι 0.5 sec και υλοποιείται με βάση τις δοσμένες συναρτήσεις της εκφώνησης. Το πρόγραμμα μας εκτελεί διαδοχικές ολισθήσεις σε έναν καταχωρητή τον οποίο αναπαριστά στα leds. Εάν το LSB της θύρας Β βρεθεί ON η κίνηση του led σταματάει προσωρινά.

**2η Άσκηση**

.include "m16def.inc"

.def input = r16

.def output = r17

.def register = r18

.def leds = r19

.def delay = r20

.def number = r21

.def flag = r22

main:

ldi register, low(RAMEND) ;initialize stack pointer

out SPL, register

ldi register, high(RAMEND)

out SPH, register

ser register

out DDRA, register ;initialize PORTA for output

clr register

out DDRB, register ;initialize PORTB for input

ldi number, 100 ;set 100 seconds delay

clr flag

flash:

in input, PINB

cp input,flag

breq label1 ;if PINB=0 we turn off the leds

mov output, input

andi input,0x0F ;use mask to keep 4 LSBs

cp input,flag ;if 4 LSBs are off we turn off the leds

breq label1

andi output,0xF0 ;use mask to keep 4MSBs

lsr output ;rotate 4 times to take the right number

lsr output

lsr output

lsr output

cp output,flag

breq label2 ;if 4 MSBs are on we turn on the leds

rcall on ;turn on leds

mov delay,input

rcall delay\_routine ;calculate delay ON and wait

rcall off ;turn off leds

mov delay,output

rcall delay\_routine ;calculate delay OFF and wait

rjmp flash

label1:

rcall off

rjmp flash

label2:

rcall on

rjmp flash

delay\_routine: ;calculate delay

mul delay,number

mov r24,r0

mov r25,r1

rcall wait\_msec

ret

wait\_usec:

sbiw r24 ,1 ;2 cycles (0.250 µsec)

nop ;1 cycle (0.125 µsec)

nop ;1 cycle (0.125 µsec)

nop ;1 cycle (0.125 µsec)

nop ;1 cycle (0.125 µsec)

brne wait\_usec ;1 ? 2 cycles (0.125 ? 0.250 µsec

ret ;4 cycles (0.500 µsec)

wait\_msec:

push r24 ;2 cycles (0.250 µsec)

push r25 ;2 cycles

ldi r24,low(998) ;1 cycle (0.125 µsec)

ldi r25,high(998) ;1 cycle (0.125 µsec)

rcall wait\_usec ;3 cycles (0.375 µsec)

pop r25 ;2 cycles (0.250 µsec)

pop r24 ;2 cycles

sbiw r24 , 1 ;2 cycles

brne wait\_msec ;1 ? 2 cycles (0.125 ? 0.250 µsec)

ret ;4 cycles (0.500 µsec)

on:

ser leds ;turn on leds

out PORTA,leds

ret

off: ;turn off leds

clr leds

out PORTA,leds

ret

Στην παραπάνω άσκηση υλοποιούμε το άναμμα και το σβήσιμο όλων των leds της θύρας εξόδου Α με καθυστέρηση που δίνεται από τα pins της θύρας εισόδου Β. Αρχικά απομονώνουμε τις δύο καθυστερήσεις που δίνονται στα 4 LSB και στα 4 MSB της θύρας Α με τη βοήθεια μάσκας, στη συνέχεια εκτελούμε τον δοσμένο τύπο και κατόπιν καλούμε τις έτοιμες συναρτήσεις που μας δίνονται από την εκφώνηση.

**3η Άσκηση**

#include <avr/io.h>

#include <stdio.h>

unsigned char shift\_left(unsigned char x);

unsigned char shift\_right(unsigned char y);

int main(void)

{

unsigned char z0,z1,z2,z3,z4,a;

int sw0 = 0,sw1 = 0,sw2 = 0,sw3 = 0,sw4 = 0;

DDRB = 0xFF; /\* PORTB as output \*/

DDRD = 0x00; /\* PORTD as input \*/

PORTB = 0x01; /\* we start with right LED on \*/

while (1)

{

z0 = PIND & 0x01; /\* we take each of the 5 LSBs of PIND by applying masks \*/

z1 = PIND & 0x02;

z2 = PIND & 0x04;

z3 = PIND & 0x08;

z4 = PIND & 0x10;

if(sw0 == 1 && z0 == 0x00) /\* if current PIN went from 1 to 0 we call a shift function \*/

{ /\* with giving priority \*/

PORTB = shift\_left(PINB);

}

else if(sw1 == 1 && z1 == 0x00)

{

PORTB = shift\_right(PINB);

}

else if(sw2 == 1 && z2 == 0x00)

{

a = shift\_left(PINB);

PORTB = shift\_left(a);

}

else if(sw3 == 1 && z3 == 0x00)

{

a = shift\_right(PINB);

PORTB = shift\_right(a);

}

else if(sw4 == 1 && z4 == 0x00)

{

PORTB = 0x01;

}

if(z0 == 0x01)

sw0 = 1;

else

sw0 = 0;

if(z1 == 0x02)

sw1 = 1;

else

sw1 = 0;

if(z2 == 0x04)

sw2 = 1;

else

sw2 = 0;

if(z3 == 0x08)

sw3 = 1;

else

sw3 = 0;

if(z4 == 0x10)

sw4 = 1;

else

sw4 = 0;

}

return 0;

}

unsigned char shift\_left(unsigned char x){

if(x == 0x80) return 0x01;

else return (x << 1);

}

unsigned char shift\_right(unsigned char y){

if(y == 0x01) return 0x80;

else return (y >> 1);

}

Στην 3η άσκηση υλοποιούμε την κίνηση ενός led δεξιά ή αριστερά, μία ή δύο θέσεις, ανάλογα με τα pins προτεραιότητας της θύρας Β. Οι αλλαγές πραγματοποιούνται αφήνοντας τα push buttons B και για το λόγο αυτό χρησιμοποιούμε αντίστοιχες σημαίες. Οι ολισθήσεις πραγματοποιούνται ανάλογα με την προτεραιότητα της άσκησης καλώντας ειδικές συναρτήσεις.